ELECT + +: Faster Conformational Search Method for Docking Flexible Molecules Using Molecular Similarity

SHINGO MAKINO,[1,2] IRWIN D. KUNTZ[1]

[1]*Department of Pharmaceutical Chemistry, School of Pharmacy, University of California, San Francisco, California 94143-0446*
[2]*Central Research Lab., Ajinomoto Co., Inc., Kawasaki, Japan*

Received 26 December 1997; accepted 29 July 1998

ABSTRACT: We have developed a program, ELECT + + (Effective LEssening of Conformations by Template molecules in C + +), to speed up the conformational search for small flexible molecules using the similar property principle. We apply this principle to molecular shape and, importantly, to molecular flexibility. After molecules in a database are clustered according to flexibility and shape (FCLUST + +), additional reagents are generated to screen the conformational space of molecules in each cluster (TEMPLATE + +). We call these representative reagents of each cluster *template reagents*. Template reagents and clustered reagents produce, after reaction, template molecules and clustered molecules, respectively (tREACT + +). The conformations of a template molecule are searched in the context of a macromolecular target. Acceptable conformational choices are then applied to all molecules in its cluster, thus effectively biasing conformational space to speed up conformational searches (tSEARCH + +). In our incremental search method, it is necessary to calculate the root-mean-square deviations (RMSD) matrix of distances between different conformations of the same molecule to reduce the number of conformations. Instead of calculating the RMSD matrix for all molecules in a cluster, the RMSD matrix of a template molecule is chosen as a reference and applied to all the molecules in its cluster. We demonstrate that FCLUST + + clusters the primary amine reagents from the *Available Chemicals Directory* (ACD) successfully. The program tSEARCH + + was applied to dihydrofolate reductase with virtual molecules generated by tREACT + + using clustered

Correspondence to: I. D. Kuntz; e-mail: kuntz@cgl.ucsf.edu
Contract/grant sponsor: General Medical Sciences, National Institutes of Health; contract/grant numbers: GM-31497, GM-39552

CCC 0192-8651 / 98 / 161834-19

primary amine reagents. The conformational search by the program tSEARCH++ was about 4.8 times faster than by SEARCH++, with an acceptable range of errors. © 1998 John Wiley & Sons, Inc. J Comput Chem 19: 1834–1852, 1998

Keywords: cluster; flexibility; similarity; docking; reaction; conformation

Introduction

Combinatorial chemistry and high-throughput screening methods have provided the ability to screen a large number of molecules in a short time.[1–5] Because the number of synthetically accessible molecules can be enormous, methods are required to cluster potential candidates to select as few molecules as possible for synthesis and assays.[6–8] However, because small differences in molecular structures may cause drastic changes in affinities of molecules for macromolecules, selection of such representative molecules might fail to find the molecules with greatest affinity. Although it is usually impractical to screen all possible molecules experimentally, some computer programs are designed to screen a very large number of combinatorially generated molecules effectively. In previous work, the programs PRO_LIGAND[9] and CombiDOCK[10] treated the conformational space of each fragment independently to reduce the amount of computation. On the other hand, DREAM++[2] works when fragments are hierarchically dependent and cannot be treated independently. Given the computational bottleneck caused by conformational searching, there is always a need for further acceleration of the search process. For this reason, we have developed the set of programs, ELECT++, to speed up conformational search by taking advantage of the similar property principle. If the flexibility and shape of two molecules are similar, then these molecules should have similar conformational constraints when bound to a macromolecule. Therefore, the conformational search of the first molecule should be able to help the conformational search of the second molecule. To apply this principle for effective conformational searching, it is necessary to cluster molecules prior to the conformational search. Instead of clustering entire molecules, we cluster the reagents that generate the molecules. This procedure reduces the complexity of the clusters and can be done at a very early stage. The clustering is performed based on reactivity, flexibility, and shape. There are many methods to describe the shape of molecules.[11,12] We have chosen to use *structural keys*[6,13] for clustering reagents, because the shapes of small molecules are significantly chemically restricted. We choose a ''basis set'' of key reagents to be used as structural keys for clustering. An advantage of using key reagents is that the information for reactivity and flexibility can be added to them. For the next step, representatives from each cluster, which we call *template reagents*, are generated to screen the conformational space of molecules in their clusters effectively. In this article, we describe the aforementioned method in detail and apply it to a system in which commercially available molecules are used. The clustering algorithm, itself, and the efficiency and accuracy of the new search method are discussed. The clustered molecules are generated using reactions between a reactant molecule and clustered molecules using the program tREACT++, which is a different version of REACT++[2] for clustered molecules.

Methods

All the molecular modelings were performed using the program SYBYL from Tripos.[14] We use a modified mol2 file format[15] with conformation and reaction information for input and output of molecules. This file format can be read by the program SYBYL.

CLUSTERING REAGENTS BASED ON REACTIVITY, FLEXIBILITY, AND SHAPE: FCLUST++

The program FCLUST++ (Flexible CLUSTering in C++) has been developed to cluster molecules especially for the new search method. Because the clustered molecules are used for reacting with an anchor molecule, we call these molecules ''reagents.'' However, the clustering procedure is general and can be applied to any molecule. We use the reagents in Figure 1 to present the algorithm. First, we describe conditions

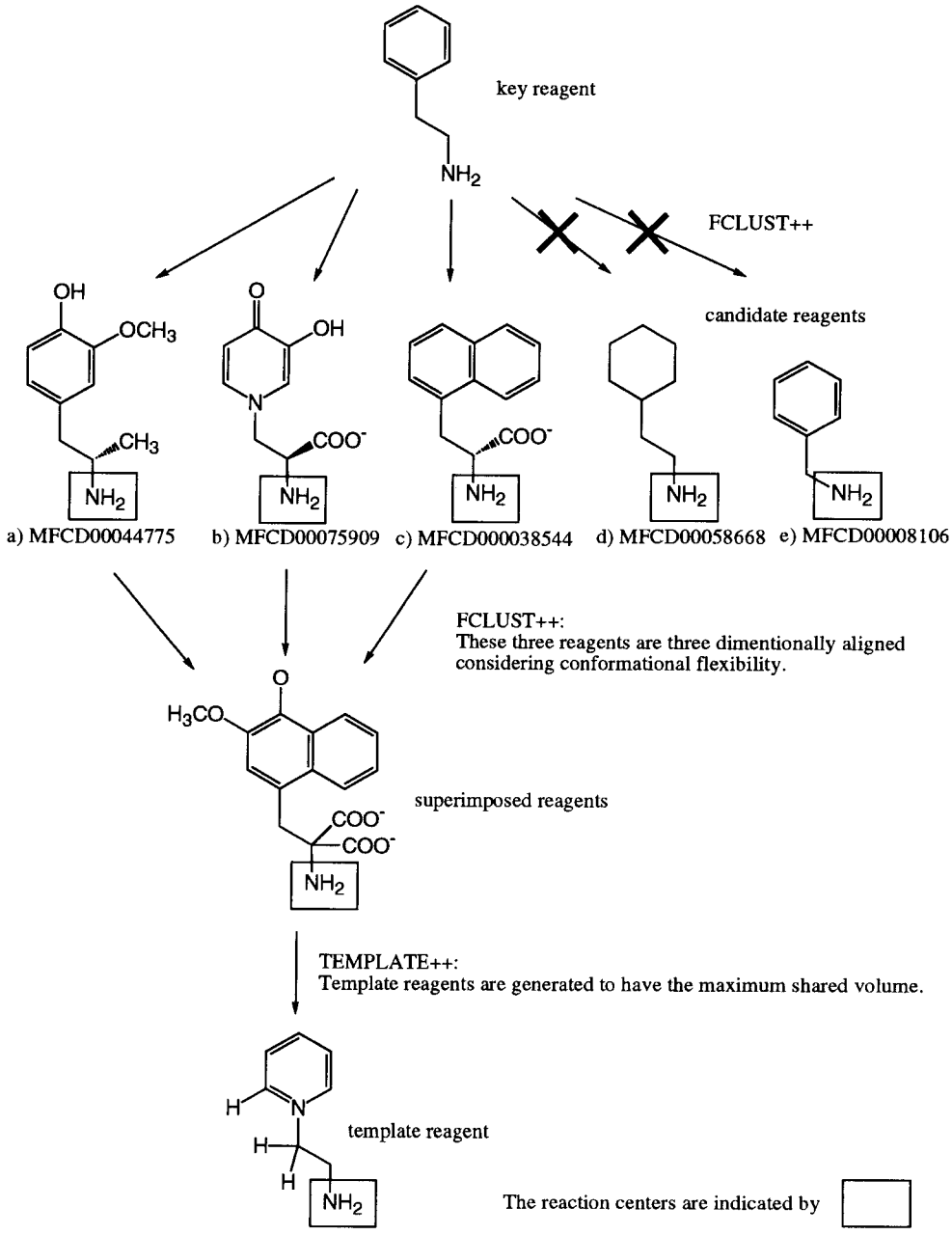


FIGURE 1. Methods for clustering and generation of template reagents.

that a candidate and a key reagent must satisfy to be matched. Then, the method to generate proper key reagents is described. A proper set of key reagents is crucial for producing appropriate clusters.

MATCHING CONDITIONS

The conditions to determine whether or not a candidate reagent matches a key reagent are: (1) reactivity; (2) flexibility; and (3) shape. These matching conditions are unsymmetrical when comparing a candidate and a key reagent as we will explain in this section. The matching is carried out at the atomic level. To include a larger set of molecules, we ignore ''implicit'' hydrogen atoms. Only those hydrogen atoms explicitly included in the molecular structures are considered for matching. For example, only the hydrogen atoms in an amino group are attached to the key reagent in Figure 1. The matching algorithm is as follows: First, all the atoms in a key reagent are mapped

onto the atoms of a candidate reagent using the exhaustive clique detection method,[16] assuring that connectivity between sets of matched atoms is always the same. In the reaction center all atoms must match each other in type. Outside this center, any atom type matches any other. A candidate reagent is discarded if any atom of a key reagent is not mapped onto a candidate reagent. The bond *types* do not have to be identical to be matched. As an example, the bond types in the ring systems of the key reagent and candidate in Figure 1b are different, but these two reagents can be superimposed on each other.

For the next step, the flexibility matching is examined between a key and a candidate reagent. Because these clustered reagents will be used in the incremental search method,[2, 17, 18, 23] the flexibility matching has to be carefully considered. It is important to note that the flexibility of a key reagent and a candidate reagent do not have to be exactly the same. The matching conditions for flexibility are shown in Figure 2. Additional flexible bonds in a candidate reagent are allowed if they can be searched by the additional incremental search method (Fig. 2a). This allowance significantly increases the number of reagents that can be clustered around each key reagent. However, the additional flexible bonds in a candidate reagent cannot be inserted between other flexible bonds of a key reagent, because these additional flexible bonds cannot be searched in the incremental search method (Fig. 2b). Furthermore, all flexible bonds in a key reagent must be flexible in a candidate reagent, so that the conformational space of a key reagent is a proper subset of the conformational space of reagents in its cluster (Fig. 2c).

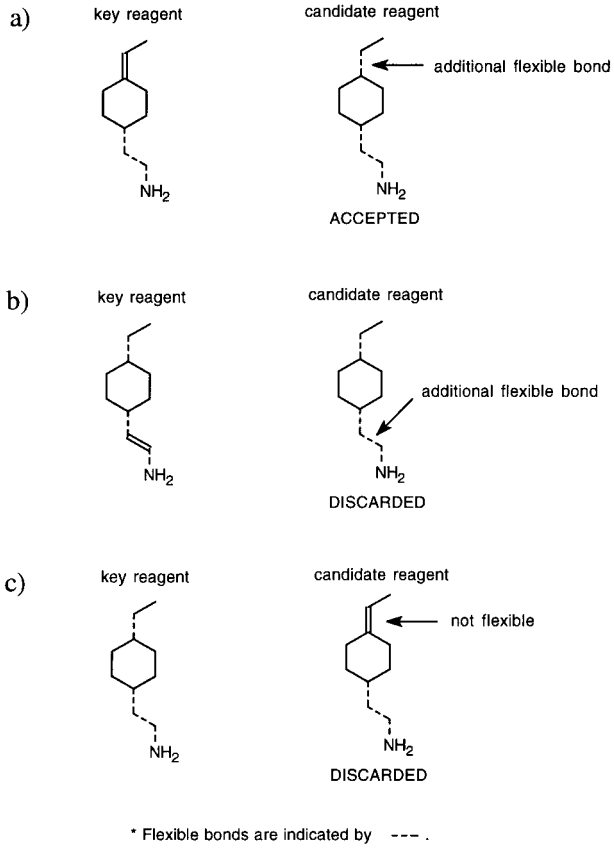
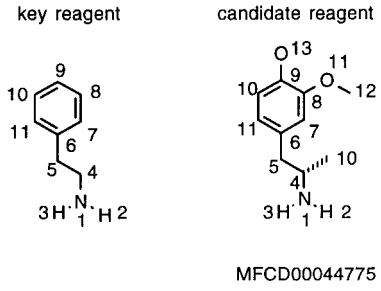


FIGURE 2. Matching condition for flexibility.

If a candidate reagent satisfies all the conditions just described, the shape matching will be tested by a three-dimensional superimposition of a candidate onto a key reagent. Because key and candidate reagents are both flexible and do not necessarily have the appropriate conformations for superimposition, the conformations of reagents have to be set to standard values before attempting to superimpose them. Thus, the torsion angles of all the flexible bonds are set at 120° to unify the conformations of a candidate reagent and key reagent. The four atoms for defining a torsion angle of a flexible bond in a key reagent are selected arbitrarily; however, the same set of atoms must be selected in a candidate reagent. For example, if the torsion angle of the key reagent is defined by atoms 2, 1, 4, 5, the atoms in the candidate atoms 3, 1, 4, 5 will be selected in the case of matching C or D in Figure 3 (3, 1, 4, 10 cannot be matched. In case of matching A or B, the candidate atoms will be 2, 1, 4, 5.) The program FCLUST++ will produce different conformations of a candidate reagent for each matching to superimpose the two molecules as much as possible. The superimposition of a candidate reagent onto a key reagent is performed by the Kabsh[19, 20] and Herman[21] algorithms.[22] The alignment giving the minimum RMSD for superimposed reagents is selected. If the RMSD value for this alignment is less than the criterion (default 0.15 Å), the candidate molecule is considered to be matched to the key reagent. Molecules with sterically different ring systems (Fig. 1d should be discarded due to three-dimensional differences) or with different chirality can be detected at this stage.

METHOD TO GENERATE KEY REAGENTS

In the previous section, we defined the matching conditions that a candidate and a key reagent have to satisfy. However, we did not describe one of the most critical points for this clustering algo-



The possible alignments

key index	candidate index (matching)			
	A	B	C	D
1	1	1	1	1
2	2	2	3	3
3	3	3	2	2
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	11	7	11
8	8	10	8	10
9	9	9	9	9
10	10	8	10	8
11	11	7	11	7

FIGURE 3. Definition of torsion angles to unify conformations. A–D are the possible mappings of a key reagent onto a candidate reagent based on the connectivity of the atoms.

rithm; that is, how to generate the key reagents. To explain the algorithm to generate key reagents, we selected a set of primary amines from ACD (Fig. 4). Because a key reagent has the same, or simpler, connectivity as any other reagents in its cluster, the following four-part algorithm can be applied: (1) The reagents in a database are sorted according to molecular weight. (2) The reagent with the minimum molecular weight is picked. A key reagent is produced by removing all the hydrogen atoms in this selected molecule except for the hydrogen atoms in the reaction center. (3) All reagents that can be clustered around this key reagent are selected and removed from the database. (4) The same procedures, (2)–(3), are repeated until all the reagents in the database are removed.

This clustering algorithm worked effectively for this database (Fig. 5). However, several potential problems can be noticed: (1) If the database does not contain any of the reagents 1, 2, 4, 5, each cluster will contain only one reagent as the result of this clustering algorithm. Thus, such critical reagents have to exist in a database for this algorithm to be successful. (2) Some clusters can be too large (cluster 2 in Fig. 5) and further division of the clusters might be preferable. The first problem is not likely to arise when this algorithm is applied to a large database. However, problem (2) may be serious, especially for a large database. To avoid this difficulty, a second generation of clusters can be developed, using additional key reagents (Fig. 6). The proper additional reagents can be

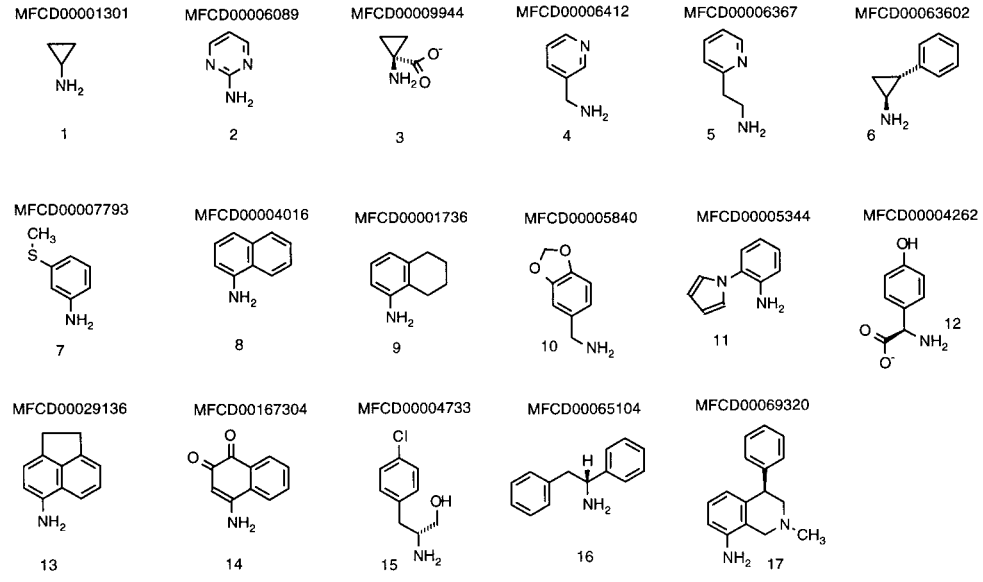


FIGURE 4. A small database from the ACD to illustrate the clustering method. The primary amino groups are assigned as the reaction center.

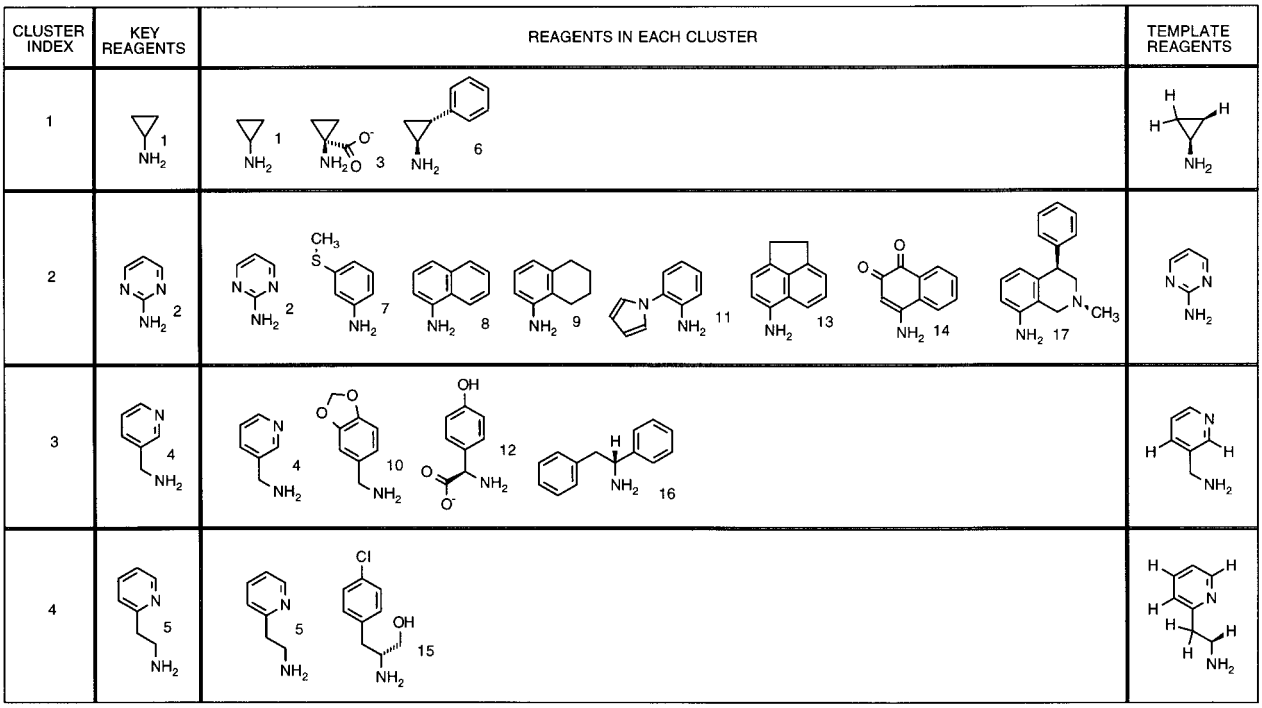


FIGURE 5. Clustering according to key reagents from the database.

estimated easily by examining the superimposed reagents of each cluster graphically. These additional key reagents are sorted according to molecular weights, from heavier to lighter, which is the opposite direction to sorting of the first database, and used prior to key reagents generated from the database. The order of key reagents is critical. For example, if key reagent 4 appears prior to molecule 1 and 2, key reagent 4 will take all the molecules in clusters 1 and 2. These potential problems will be discussed in the Experimental section using a real database.

GENERATION OF TEMPLATE REAGENT FROM REAGENTS IN A CLUSTER: TEMPLATE++

The representative reagents of each cluster are called *template reagents* in this study. Template reagents represent a certain reactivity, flexibility, and shape of reagents in each cluster. Although key reagents themselves can be template reagents, we optimize them to screen conformational space effectively. To do this, we have developed the program TEMPLATE++ (Fig. 1). To generate a template reagent, we add atoms to a key reagent if they satisfy the following conditions: (1) the additional atoms exist in all molecules of the cluster; and (2) the maximum distance between these corresponding atoms is less than 0.25 Å.

The atom types in a template reagent must have a minimum Van der Waals (VdW) radius of atoms at the corresponding positions of all reagents in its cluster. Although the template reagents possess chemical structures, they are virtual molecules created to screen the conformational space of molecules in their clusters in computers. Thus, these template reagents do not have to be limited to stable molecules. The computer-generated template reagents for each cluster are shown in Figure 5 and 6. As for key reagents, implicit hydrogen atoms do not exist in template reagents throughout this article. Also, note that the clusters in Figure 6 have more complicated template reagents than those in Figure 5, because the reagents in the divided clusters are more similar to each other, making it possible to make more specialized template agents.

CONFORMATIONAL SEARCH USING TEMPLATE REAGENTS: TSEARCH++

Molecules to be screened are generated using reactions between a reactant and reagents. The products generated by reactions between a reactant and template reagents are called *template*

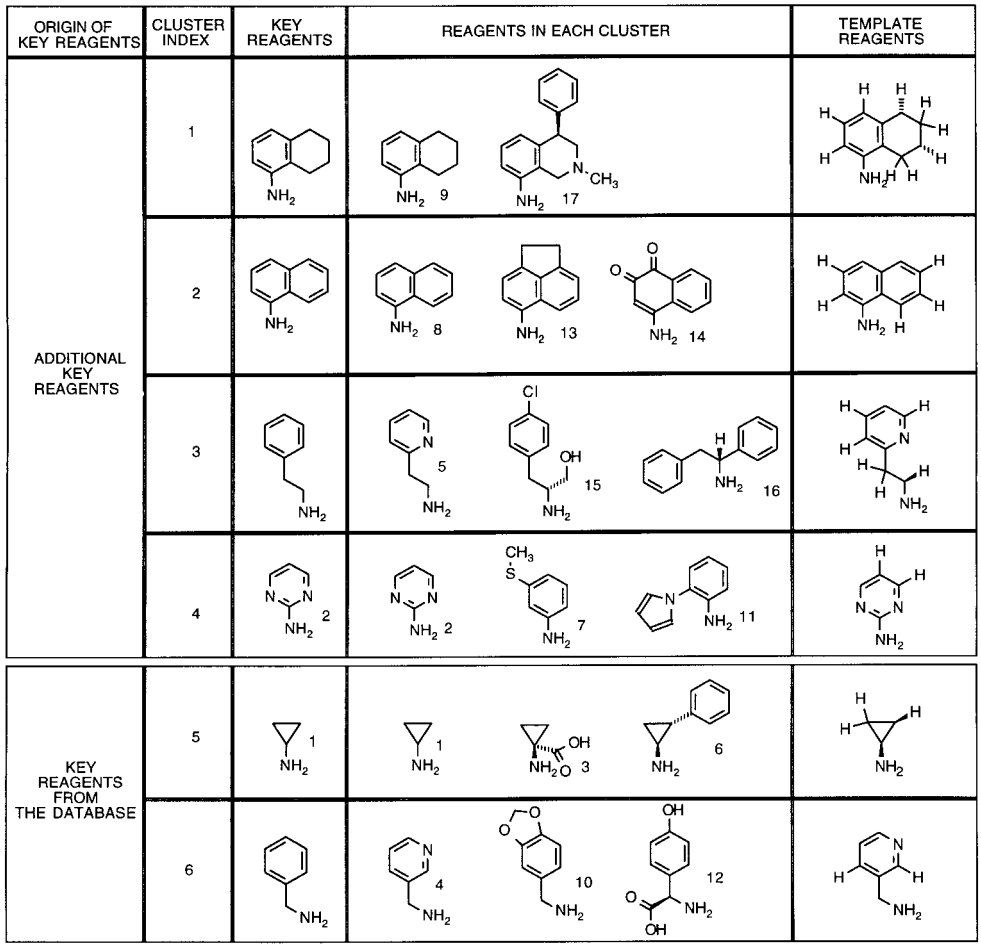
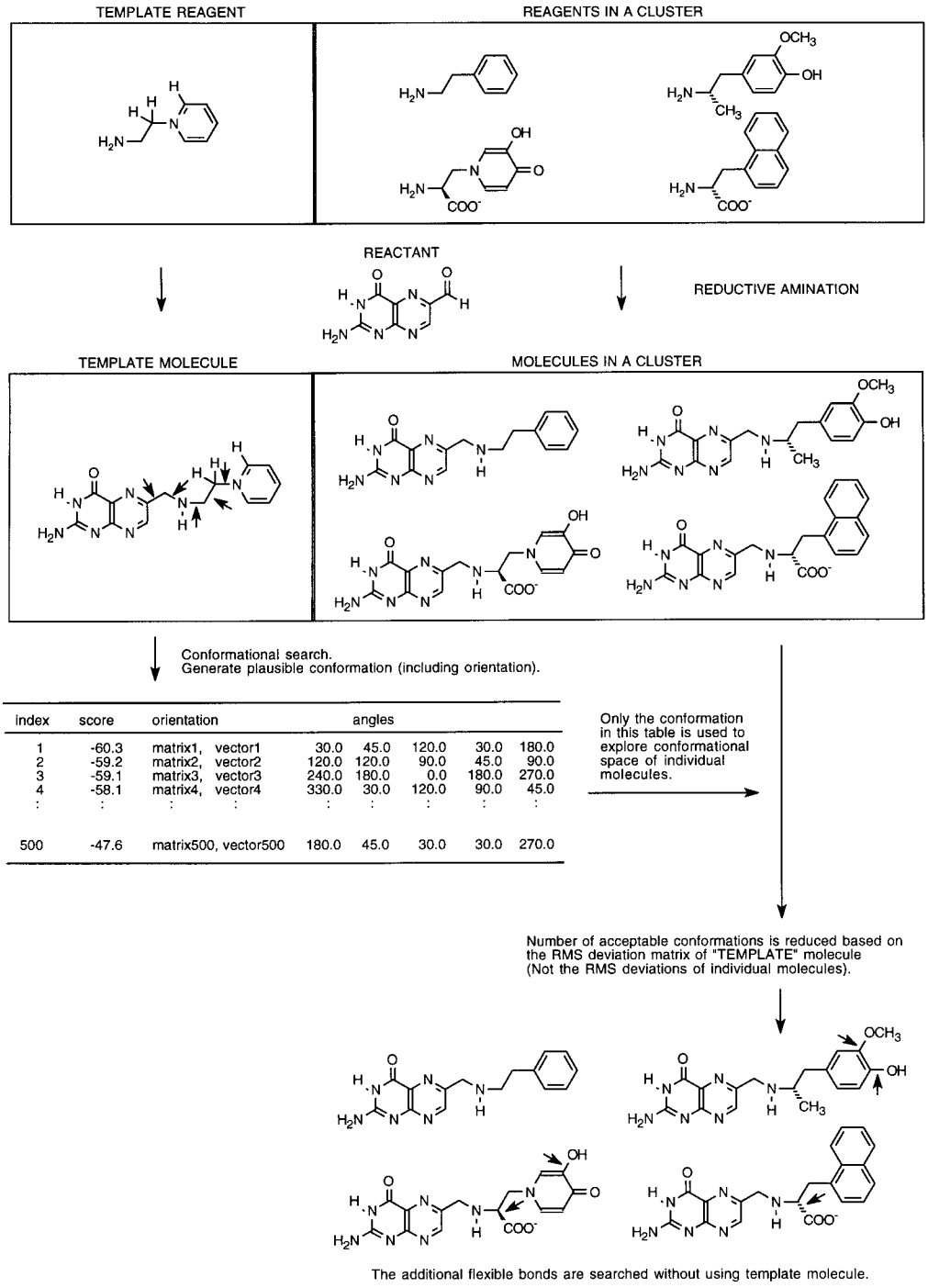


FIGURE 6. Clustering using the additional key reagents.

molecules. Sets of reagents from clusters of products. Template molecules are the representatives of these clusters of products (Fig. 7). We have developed a new program that can rapidly explore conformational space of sets of clustered molecules. The new program, tSEARCH + + (template SEARCH + +), uses the template molecules to screen the conformational space of molecules in its cluster. Only the VdW term is used for evaluating conformations of template molecules, because template reagents are not designed to represent electrostatic features of sets of reagents. The plausible conformations of a template molecule bound to a macromolecule are applied to molecules in its cluster, giving the scores for each conformation of each molecule in its cluster. Because a template molecule possesses the minimum steric features of the molecules in its cluster, the conformations that are not possible for a template molecule will never be possible for any of the molecules in its cluster, assuming that the criteria for superimpositions of the reagents are strict enough. However, the conformations possible for a template molecule are not always possible for the molecules in its cluster, because additional steric features are present in the molecules in its cluster. If there are additional flexible bonds in a molecule to be searched, then the scores of the components with these flexible bonds are not included at this stage.[23] Finally, the conformations of the additional flexible bonds are further searched without conformational prescreening in the same manner as in SEARCH + + (Fig. 7).

The RMSD matrix of all configurations of a template molecule represents the RMSD matrix of all configurations of the molecules in its cluster.



index	score	orientation	angles				
1	-60.3	matrix1, vector1	30.0	45.0	120.0	30.0	180.0
2	-59.2	matrix2, vector2	120.0	120.0	90.0	45.0	90.0
3	-59.1	matrix3, vector3	240.0	180.0	0.0	180.0	270.0
4	-58.1	matrix4, vector4	330.0	30.0	120.0	90.0	45.0
:	:	: :	:	:	:	:	:
500	-47.6	matrix500, vector500	180.0	45.0	30.0	30.0	270.0

FIGURE 7. Conformational search using template molecules.

The reduction in computation time can be achieved by: (1) the limitation of conformational space to be explored, and (2) avoiding calculations for the RMSD matrix of each molecule in a cluster. The conformational searches for the additional flexible bonds will not gain any computational advantages based on the similarity of molecules.

Because the conformational searches for molecules in a cluster are sequential, the memory size required by tSEARCH++ is not proportional to the number of molecules in a cluster. However, the required memory size is proportional to the number of conformations to be applied to each molecule in a cluster.

Results

CLUSTERING REAGENTS

We used the ACD as a source for reagents. Before clustering reagents, the molecules that seem to be inappropriate as reagents in combinatorial chemistry were removed. The molecules removed from the database were: (1) molecules with more than five flexible bonds; (2) molecules with molecular weights > 250.0 Da; and (3) molecules with peptide, ester, and sulfonamide bonds. These last molecules may be easily produced by simulating conventional chemistry in computers. These processes eliminated about 37% of the molecules from the ACD. Then, molecules with a certain functionality were chosen as the reagents for selected reactions. In this article, we focused on the primary amines as the reagents for reductive amination. Other reagents were selected and clustered in the same manner as described for the primary amines, without difficulty (data not shown).

Molecules with only one primary amino group and without a secondary amino group were selected as the reagents for reductive amination. A total of 2381 reagents satisfied this condition after removing duplications. The automatic clustering failed, producing only one large cluster, because the database contains the very simple reagent, methylamine, which is a structural part of all other molecules in the database. The production of such a large cluster was easily avoided by grouping the reagents in the database into those with and those without ring systems before clustering, producing the clusters shown in Figure 8. For the second generation of clusters, some of these clusters with a large number of reagents were further divided. The additional key reagents and the number of reagents in each cluster are shown in Figure 9. We designate the first set of clusters as *first clusters* and the second set of clusters as *second clusters* in the following sections.

EVALUATION OF SEARCH METHODS

In what follows, we compare scores and speed of calculations between SEARCH + + and tSEARCH + +. As a test system, we chose dihydrofolate reductase (DHFR) with various virtual

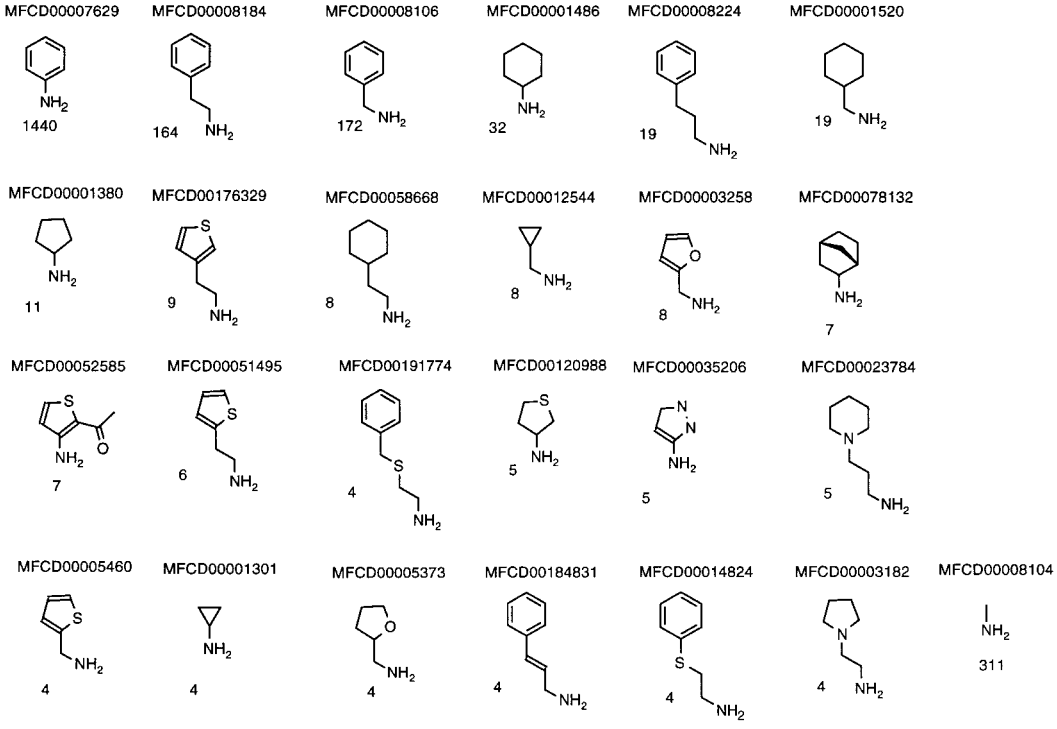


FIGURE 8. Clusters only by key reagents from the database. The number of reagents in each cluster is shown below each structure.

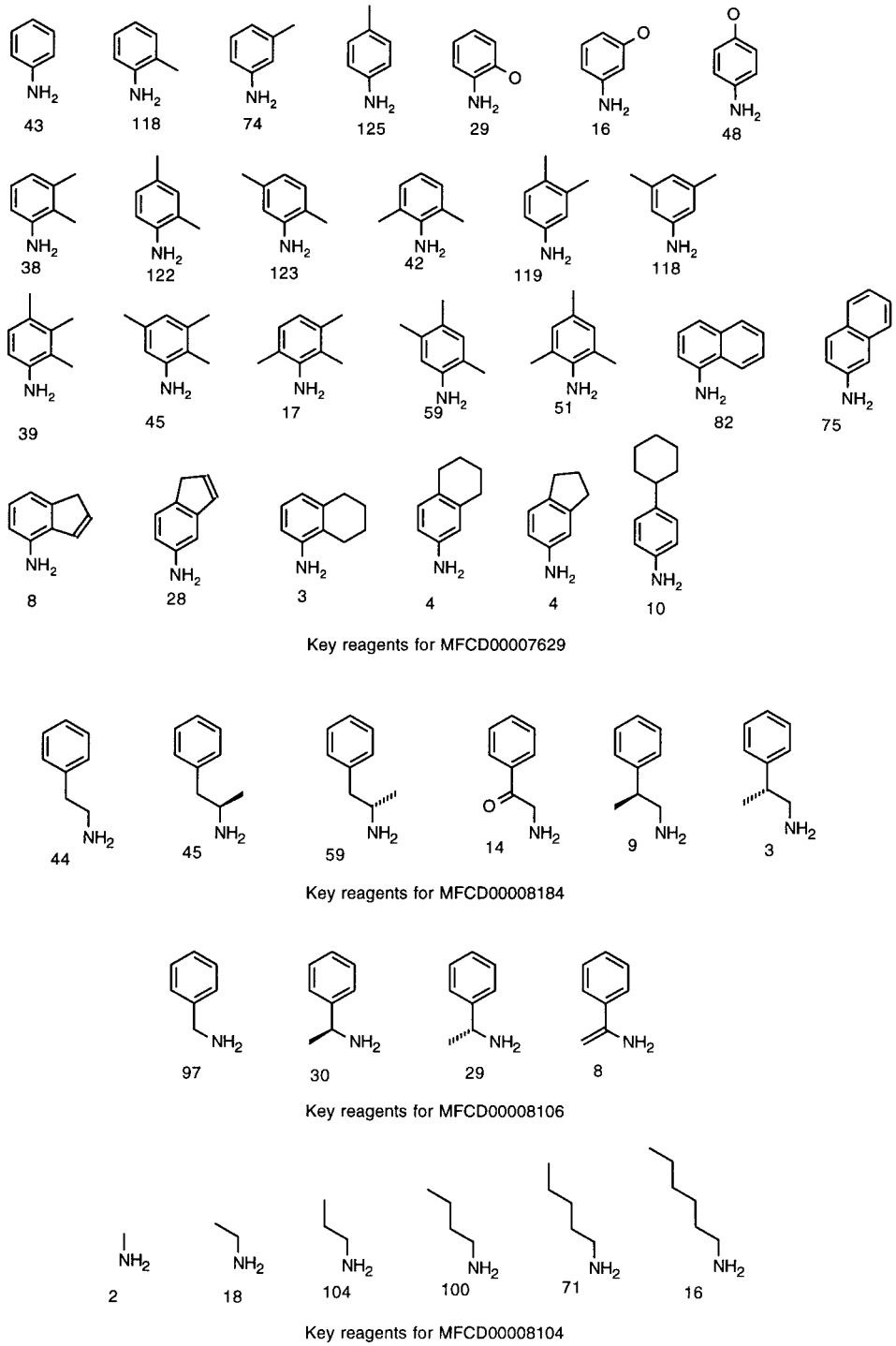


FIGURE 9. Clusters divided by using the additional key reagents. Although other clusters are slightly different due to the shift of reagents (Fig. 10), they are almost identical to clusters in Figure 8.

molecules generated by the reaction *reductive amination* (Fig. 7). Because a reactant and all reagents taken from the ACD were used for reductive amination, by which many of the folate derivatives have been synthesized,[24–28] we believe this system is applicable to real chemistry. The protein structure was taken from 7dfr[29] of the Brookhaven Protein Data Bank (PDB).[30] After removing the folate substrate from the protein complex and assigning the Gasteiger–Marsili charges,[31,32] the hy-

drogen atoms were generated and minimized, fixing the heavy atoms. The programs SYBYL was used for all of these calculations. The orientations of the anchor fragment were generated by the program ORIENT + + (min_distance 1.4, distance_tolerance 0.6).[2,23] All flexible bonds were searched in comparing the different search algorithms, using the increment of torsion angle of 20° uniformly. The maximum score for all the molecules was set at 10 kcal/mol. The method to calculate RMSD was improved from that in the previous version of SEARCH + +. All RMSD matrices of partial components calculated between different conformations of each molecule were stored with hashed indices of transformation functions, thus avoiding the repetition of RMSD calculations for any partial components. In the following sections, we do not distinguish between conformational space and configurational space, as conformation and orientation are searched together.[2]

In the programs SEARCH++ and tSEARCH++, all calculations are the same except that the following new features are used in tSEARCH + +: (1) the reduction of conformational space of each molecule in a cluster using a template reagent; and (2) the use of the RMSD matrix of a template reagent for all the molecules in its cluster to reduce the number of conformations. For evaluating the programs SEARCH + + and tSEARCH + +, we simply compared the AMBER force-field[33,34] scores[2] obtained. Our forces was adequacy of sampling rather than testing the scoring functions themselves.[35]

COMPARISON OF REACTION PROGRAMS: REACT + + AND tREACT + +

We have been developing the program REACT + +[2], which generates products from reactants and reagents through chemical reactions, whereas the partial conformations of the products are inherited from the reactants. The program transforms the original coordinates of reagents based on the atom coordinates of the reaction centers of each reactant and each reagent to connect them covalently. As a result, when the program was applied to the reactant and the superimposed reagents, they partially altered the initial alignments in the products, resulting in inappropriate alignments (Fig. 10a). On the other hand, if the transformation matrix and vector used to transform the coordinates of a template reagent are used for all reagents in a cluster, the initial alignment will be preserved through reactions (Fig. 10b). We call the program to generate products based on this algorithm tREACT + +. The average differences between the set of molecules generated by REACT + + and tREACT + + were 0.1 Å for bond length and 0.12° for bond angle around new covalent bonds generated by reaction programs. A possible problem is that a slightly different set of products might have significantly different best scores. To examine this problem, we calculated the best scores for both sets of molecules using SEARCH + +. The RMSD value for this set of scores was 1.5, thus the molecules generated by REACT + + and tREACT + + possessed similar best scores.

NUMBER OF CONFORMATIONS APPLIED TO CLUSTERED MOLECULES

Because tSEARCH + + explores a much smaller geometric space than SEARCH + +, we examined, whether the program tSEARCH + + could find equivalent scores. The conformational space explored by tSEARCH + + is directly proportional to the number of conformations applied to molecules in clusters. Therefore, the number of conformations was set to 100, 500, and 1500 to compare the best scores by SEARCH + + and tSEARCH + + (Fig. 11). In all cases, the slopes were less than 1.0, which indicates that SEARCH + + found better sets of scores, on average. If we focus on the best scoring molecules, tSEARCH + + improves, as RMSD values indicate, the result of calculation from 100 to 500 (see circle in Fig. 11a). However, there was little additional improvement from 500 to 1500.

The speed-up factor depends significantly on the number of conformations to be explored by tSEARCH + +: with 500 conformations it was 4.8. The selection of a smaller number of conformations increases the speed-up factor (e.g., the factor will be 9.5 with 100 conformations). However, the selection of even smaller conformation numbers (below 100) scarcely increases the speed-up factor; furthermore, because it is necessary to perform the conformation search of the template molecule in advance, the speed-up factor is 12.0, even if the conformation number to be searched by tSEARCH + + is 0.

One may notice that some scores obtained by tSEARCH + + were better than SEARCH + + even though tSEARCH + + explored a smaller confor-

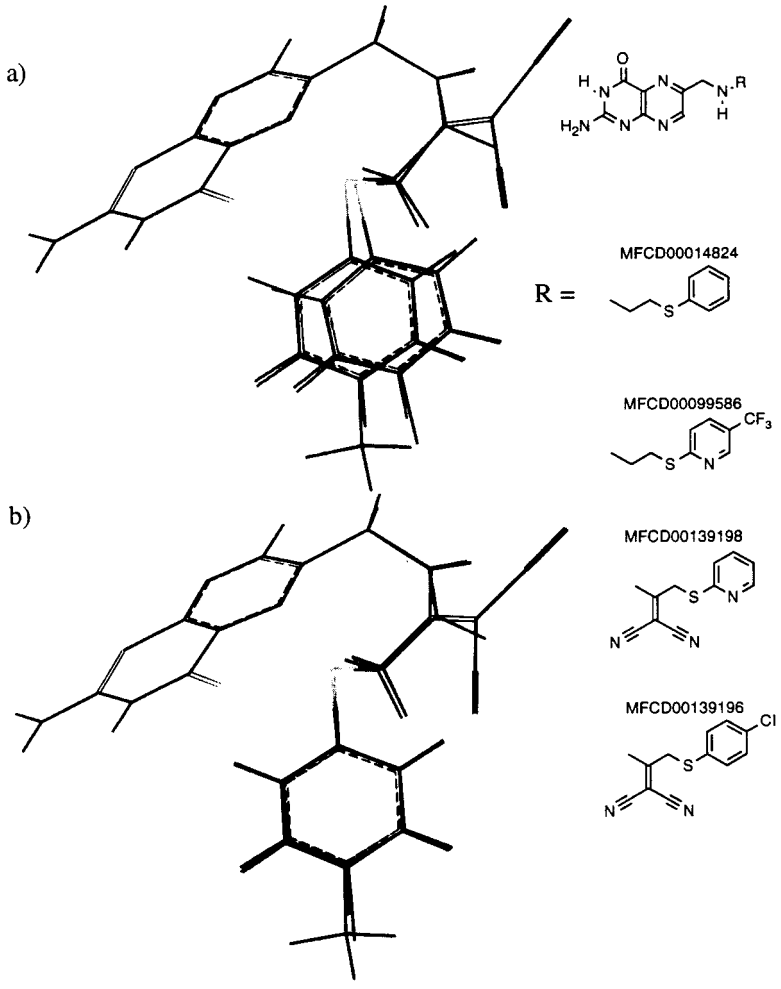


FIGURE 10. Example of clustered molecules generated by different reaction programs.

mational space. The reason is that the smaller conformational space was not exactly a subspace of the larger space because the conformational search was biased toward a certain number of conformations based on RMSD and scores.

The ability to regenerate x-ray structures is outside the scope of the present work and the x-ray structures for the molecules generated are unknown. However, most of the top-ranking molecules seem to possess plausible binding modes, similar to the binding mode of DHFR/DHF.

COMPARISON OF DIFFERENT SETS OF CLUSTERS IN SCORE AND COMPUTATION TIME

In the previous subsection, we developed two sets of clusters. The scores between the different sets of clusters were compared using tSEARCH++ to examine the appropriate size of clusters. The comparison of the sets of clusters with and without ring systems are shown separately in Figure 12. The best scores obtained in the second clusters were better in the case of ring systems (Fig. 12a), because template reagents in second clusters were more specialized and could screen conformational space more efficiently than those in first clusters. The computation time for second clusters was slightly longer (13%) than that for first clusters, due to the searches for additional template molecules. However, the results of the searches for the molecules without ring systems were the opposite, with the scores for first clusters better in most cases (Fig. 12b). Without rings the template molecules in the second clusters were much more flexible than those in first clusters. As a result, the best scores were missed by excessive reduction of conformational space using template molecules with more flexibility. However, this de-

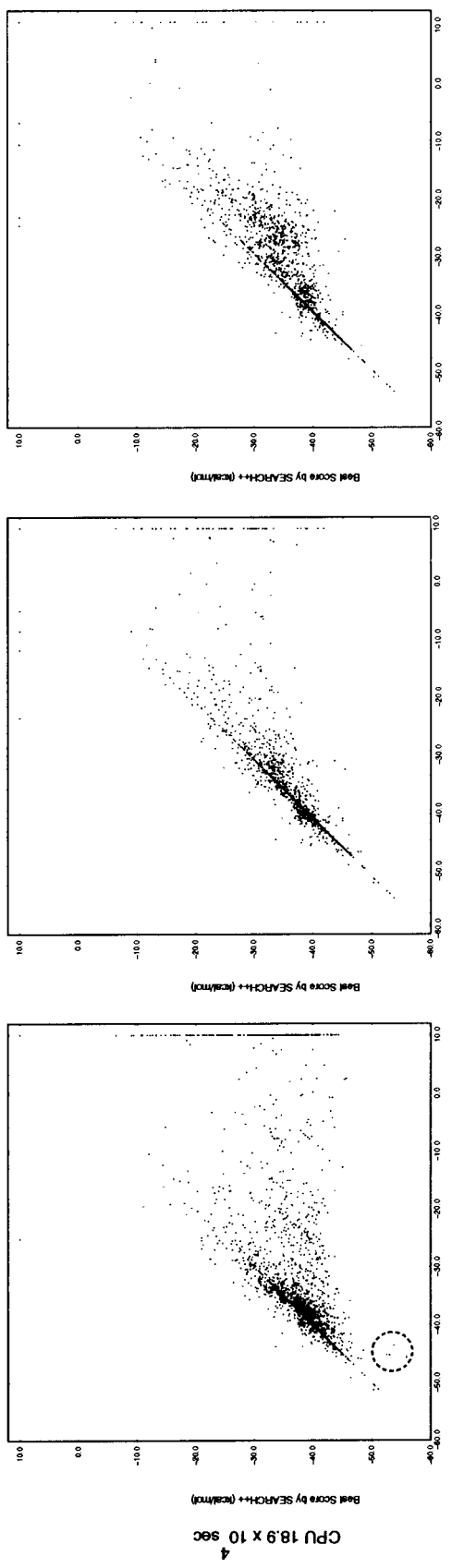


FIGURE 11. The dependency of the accuracy of the best scores on the number of conformations applied. The darkened region along the diagonal shows the very high density of molecules resulting in the same best score using the better search method.

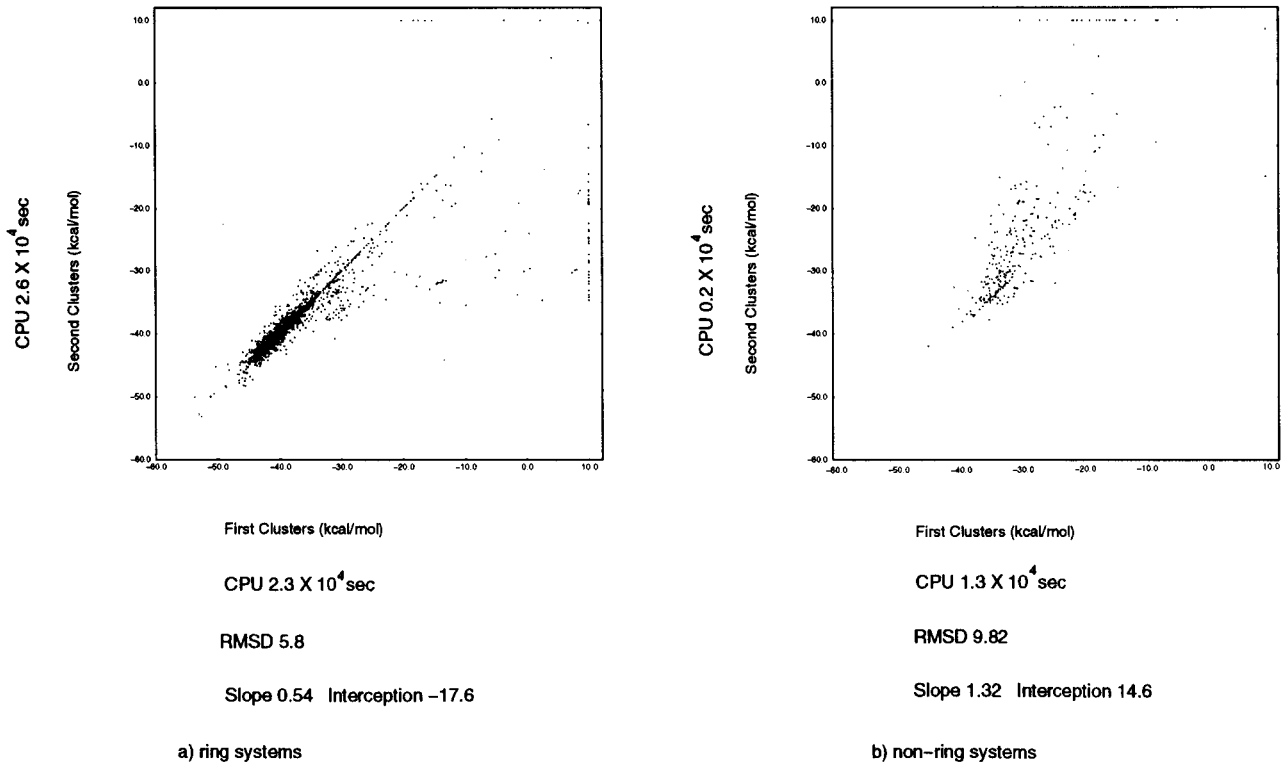


FIGURE 12. Comparison of scores between first and second clusters.

ficiency might not be serious for screening a large number of molecules and identifying high-score molecules, because the correlation of sets of best scores became better when scores were high. For example, the RMSD values for the top half and top quartile of scores of nonring systems were 2.6 and 1.8, respectively. As a result of this large conformational space reduction, the computation time for nonring systems was about 6.5 times shorter in second clusters than in first clusters.

EFFECT OF ADDITIONAL FLEXIBLE BONDS ON COMPUTATION TIME

If flexible bonds of molecules in a cluster are represented in its template molecule, the conformational space related to these flexible bonds can be effectively reduced by the template molecule. However, if molecules in a cluster have additional bonds that are not represented in its template molecule, the conformational searches for these additional flexible bonds will not gain any computational advantages from tSEARCH++. We examined how much the conformational searches for additional flexible bonds contributed to the total computation time for each molecule. The computation time in searching template molecules is not included in this analysis, because it is related to the number of molecules in each cluster, as we will explain in the following section. The computation times for each molecule are plotted based on the number of additional flexible bonds in each molecule (Fig. 13a). When the additional flexible bond was 0, the computation time was 1.9 seconds, on average. Although the computation time seemed to increase only linearly as the number of additional flexible bonds increased, this increase was significant, considering the average computation time for molecules with five additional flexible bonds took 32 times more than the average computation time for molecules with no additional flexible bonds. Therefore, the computation time in searches for additional flexible bonds comprised most of the computation time. The ratios of computation times by SEARCH++ and tSEARCH++ are plotted in Figure 13b. When there were no additional flexible bonds, the calculation time by tSEARCH++ was 43 times shorter, on average, than that by SEARCH++. However, the calculation time was only slightly shorter (1.3 times) when there were five additional flexible bonds. Although the computation time can be shortened

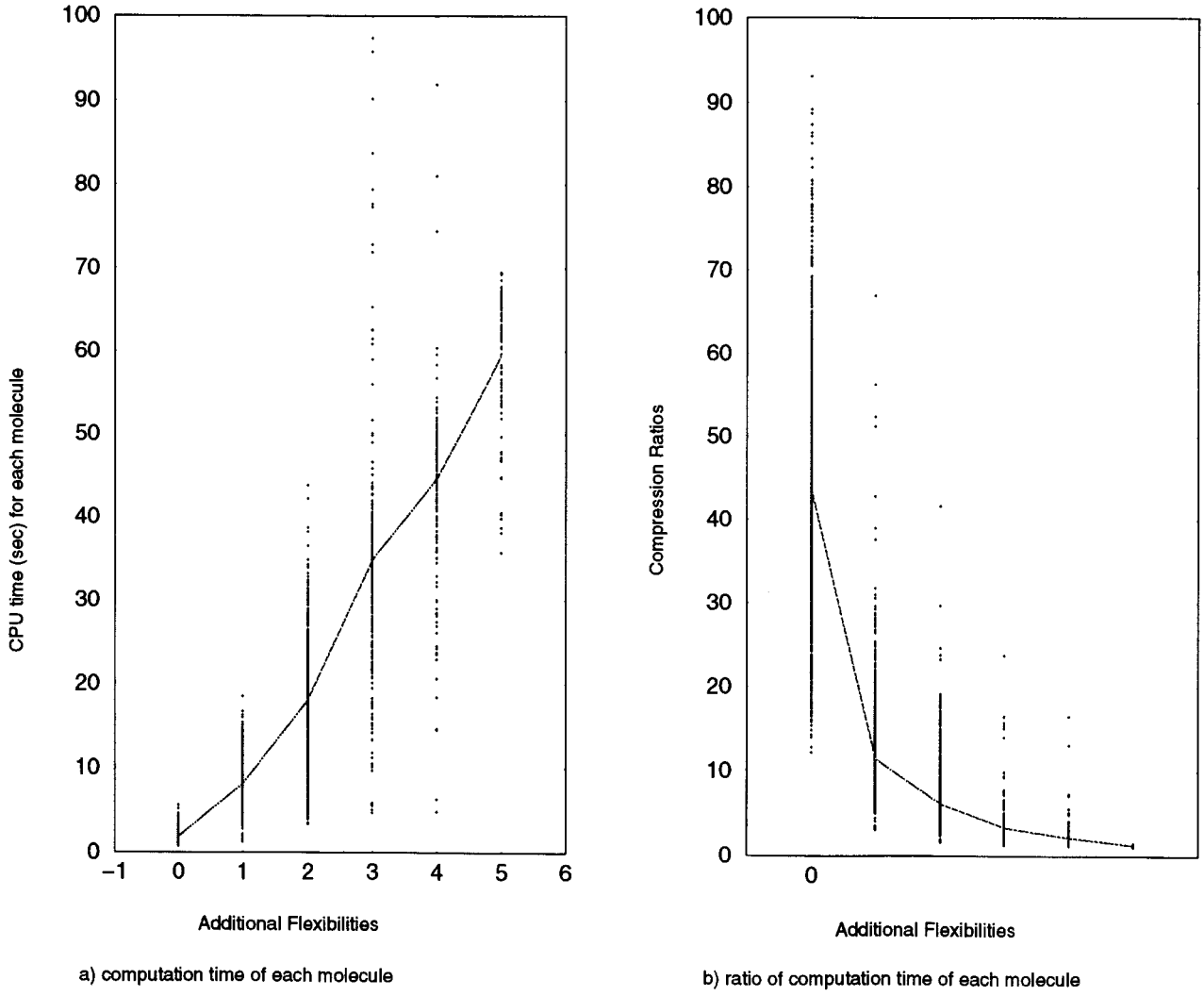


FIGURE 13. Comparison of calculation times for each additional flexibility.

by making template molecules more flexible, we must be careful about too much reduction of conformational space, as noted in the case of nonring molecules in second clusters.

EFFECT OF NUMBER OF MOLECULES IN EACH CLUSTER ON COMPUTATION TIME

In the previous section, we examined the reduction of computation time without considering the computation time for template molecules. Here, we examine the average computation time in searching molecules in each cluster, including the computation time for template molecules. The overhead time caused by conformational searches for a template molecule decreases as the number of molecules in each cluster increases, because one template molecule for each cluster has to be searched. For examining the appropriate number of molecules in a cluster, the computation time efficiency using tSEARCH + + was plotted against the number of molecules in their clusters (Fig. 14). Although the efficiency in computation time seemed to increase almost linearly until the number of molecules in each cluster is reached 10, there seemed to be no significant correlation in large clusters.

These results indicate that the division of a cluster is useful if the number of molecules in the cluster is more than ten, because screening for conformational space can be more effective with more specialized template molecules. Although it is not advisable to break up clusters when they have less than ten molecules, the calculation was

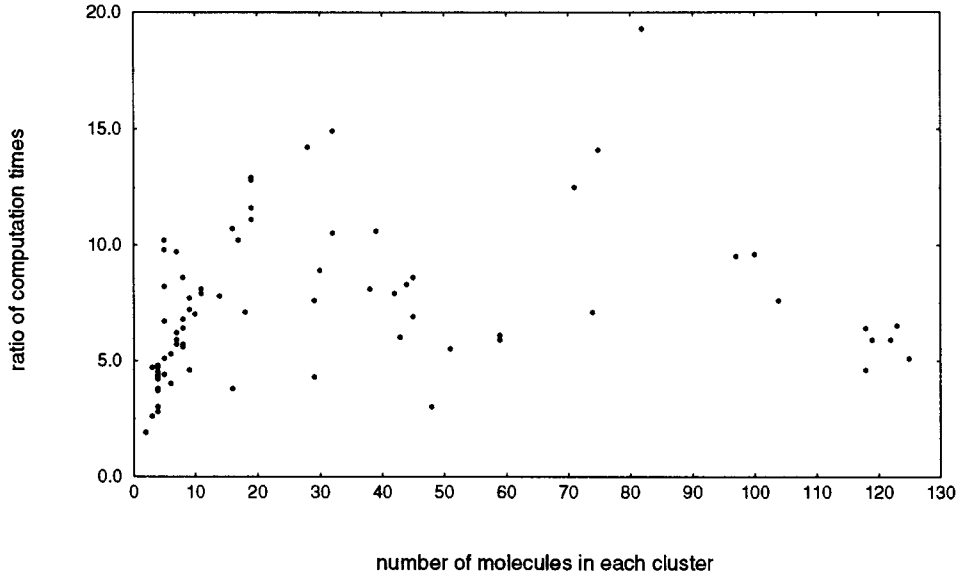


FIGURE 14. The effect of the number of molecules in each cluster on computation time.

still faster by using tSEARCH++, even if the number of molecules in a cluster was less than ten.

INCREASED EFFICIENCY OF COMPUTATION TIME IN SCREENING THE ENTIRE DATABASE

Molecules have to be clustered properly to gain computational advantages from tSEARCH++. Therefore, the entire database cannot be searched efficiently by tSEARCH++ if most molecules in the database are not clustered. For the primary amine case, 95% of reagents were grouped into the clusters with four or more molecules in them. There were 118 molecules left unclustered and were not used in the previous sections. Although these molecules could not be searched using tSEARCH++, the computation time to search these molecules had to be considered to estimate the total increase of efficiency for screening the entire database. Therefore, these nonclustered molecules were searched only by SEARCH++, giving the computation time 0.4×10^4 seconds. The total computation time was 3.7×10^4 seconds for the first clusters and 2.6×10^4 seconds for second clusters by tSEARCH++. Because the total computation time for clustered molecules by SEARCH++ was 19.3×10^4 seconds, the increase of efficiency was:

1. $(19.3 \times 10^4 + 0.4 \times 10^4)/(3.7 \times 10^4 + 0.4 \times 10^4) = 4.8$ times for first clusters.
2. $(19.3 \times 10^4 + 0.4 \times 10^4)/(2.6 \times 10^4 + 0.4 \times 10^4) = 6.6$ times for second clusters.

The better efficiency for second clusters came from the search of nonring systems, where the accuracy of the calculations was less. Therefore, we tentatively conclude that the increase of efficiency for this system was 4.8 times.

PROGRAMMING AND RESOURCE USAGE

This program is written in C++ based on a library that has been developed for DREAM++. The Standard Template Library (STL), which provides generic container classes (e.g., list, queue, binary trees), is especially useful for providing a simple interface for complicated data structures. All the programs are compiled by GNU gcc-2.7.2. All calculations were performed on Silicon Graphics Indigo 2 workstations with 200-MHz R4400 processors and 128-MB RAM.

Discussion

The ELECT++ program has three critical parts: the clustering section; the reaction section; and the conformation searching. For each part, we now describe some potential improvements and additional uses of the basic algorithms.

Because the clustering step is a preprocess, and is done only once for a database, we did not optimize the speed of this step. However, the use of a structural key should improve the process by prescreening candidate reagents. The clustering method we used has a dependency on the order of key reagents, because reagents will be clustered around the first key reagent for which all matching conditions are satisfied. As a result, there is only a single copy of any reagent throughout the clusters, thus avoiding repetition of conformational search of the same molecule. However, one might prefer to screen only some of clusters instead of screening all molecules in a database when the type of molecules can be biased. In that case, reagents should be put into all possible clusters so that any cluster contains all the reagents that match it. There is no order dependency on key molecules in this case. However, the search program must be modified to identify molecules that have already been searched to avoid repetition of the conformational search of the same molecule.

The clustering algorithm we employed was fundamentally simple: pick all the reagents that match a key reagent, where key reagents define the center of each cluster. This algorithm is similar to the cell-based partitioning method, even though the distances between key reagents are not considered. The merit of the cell-based partitioning method is that only the distances between centroids and objects have to be considered, thus avoiding calculations of distances between objects. It is necessary to define additional centroids (key reagents) to break down clusters to desirable sizes.

In addition, the clustering method can be useful for optimizing a lead compound, especially when the binding mode of the lead compound is unknown. Because all molecules in a cluster have the same basic flexibility and shape, the assumption of specific binding modes can be unnecessary for replacing a fragment of the lead compound by other fragments in the same cluster. Of course, the same topological flexibility does not guarantee that the molecules can have equivalent conformations because of the differences in the internal structure constraints in each molecule.

As we mentioned in the Results section, a template reagent was used to transform the coordinates of reagents to form covalent bonds with a reactant so that the initial alignments of reagents were not altered (tREACT + +). The alteration of the alignments can be reduced if the coordinates of the atoms in reaction centers are weighed when reagents are superimposed by the Kabsh–Herman algorithm at the clustering stage. In that case, not only the RMSD of distances, but also the deviations of distances, have to be examined. However, such weighted superimposition will reduce the number of reagents that are potential members of each cluster.

The search algorithm used in tSEARCH + + can coexist with other algorithms developed for improving the calculations for docking molecules that are synthesized by combinatorial chemistry. Such algorithms used in previous reports include: (1) the algorithm to inherit conformations through reactions,[2] which was also used in this article; and (2) the programs PRO_LIGAND and CombiDOCK, which made calculations faster by searching the conformations of each fragment independently. The new method speeds up the calculation for conformational searches of *each fragment* based on similarity of molecules. Therefore, this method can be implemented and used together with the other methods referred to in (1) and (2).

Although we used the similar property principle only in the conformational search algorithm, the same principle can be applied to the DOCK algorithm[11,36] for searching orientational space. In that case, template molecules are docked to screen the orientational space of molecules in their clusters. Because the atoms in a molecule are used to guide the docking orientations (sphere points), and the set of atoms in a template molecule is a subset of the set of atoms in any molecule in its cluster, the orientational space explored by a template molecule will be only a subset of the space of each molecule in its cluster. (It is not strictly the subset, because small deviations of distances are allowed in alignment of molecules.) For this reason, the conformational space to be explored by the template molecules is reduced by template molecules themselves. This is different from the reduction of orientational space *after* searching the orientational space using template molecules. Therefore, the application of the similarity searching method to the DOCK algorithm might not be as successful as the application to the conformational search algorithm.

We accelerated the conformational search calculations in tSEARCH + + by reducing the conformational space to be explored for molecules in clusters. However, the reduction of search space has to be compromised with accuracy of calculation, because excess reduction causes the search program to miss important conformations. The amount of reduction of conformational space is

dependent on: (1) the number of conformations applied to each molecule in clusters; and (2) the size of the conformational space of template molecules. The reason for factor 1 is that the magnitude of reduction is directly proportional to the number of possible conformations discarded if the number of possible conformations is fixed. On the other hand, the reason for factor (2) is that the reduction ratio is inversely proportional to the size of conformational space if the number of applied conformations is fixed. The size of the conformational space is increased exponentially by the number of flexible bonds in template molecules. The reason why the calculations for nonring systems in second clusters are faster but inaccurate is due to the large reduction of conformational space according to factor (2). This excess inaccuracy can be circumvented by changing the number of conformations according to the size of the conformational space. The size of increment for torsion angles in a conformational search is a different issue. It will change the resolution in conformational searches for template molecules; however, it does not directly change the reduction ratio of the conformational space of each molecule in its cluster.

Conclusion

We have developed the set of programs ELECT++ to assist the new search algorithm. The process proceeds as follows: (1) The program FCLUST++ successfully clusters the primary amine reagents from ACD according to reactivity, flexibility, and shape. (2) The program tREACT++ produces clustered molecules, based on a reaction using clustered reagents without changing the alignments of the reagents. (3) The program tSEARCH++ accelerates the conformational search using the similar property principle. tSEARCH++ can search all molecules in the database about six times faster than SEARCH++ in the acceptable range of errors. The reduction of computation time is due to the appropriate bias toward conformational space and the substitutions of the RMSD matrix of each molecule in clusters by that of template molecules.

Acknowledgments

We thank Dr. Todd Ewing for many useful discussions and suggestions. We also thank Greg Couch of the UCSF Computer Graphic Laboratory for helpful suggestions. Tripos Associates provided the SYBYL program, and Molecular Design, Ltd., provided the *Available Chemicals Directory*, for which we express our appreciation.

References

1. M. Johnson and G. M. Maggiora, In *Concepts and Applications of Molecular Similarity*, Wiley, New York, 1990.
2. S. Makino, T. J. A. Ewing, A. G. Skillman, and I. D. Kuntz, *J. Comput.-Aided Mol. Design* (accepted for publication).
3. *Available Chemicals Directory*, distributed by Molecular Design, Ltd., San Leandro, CA.
4. L. A. Thompson and J. A. Ellman, *Chem. Rev.*, **96,** 555 (1996).
5. P. H. H. Hermkens, H. C. J. Ottenheijm, and D. Rees, *Tetrahedron*, **52,** 4527 (1996).
6. R. D. Brown and Y. C. Martin, *J. Chem. Inf. Comput. Sci.*, **36,** 572 (1996).
7. *DiverseSolutions Version* 2.0.1, Laboratory for Molecular Graphics and Theoretical Modeling, College of Pharmacy, University of Texas, Austin, TX.
8. D. J. Cummins and C. W. Andrew, J. A. Bentley, and M. Cory, *J. Chem. Inf. Comput. Sci.*, **36,** 750 (1996).
9. C. W. Murray, D. E. Clark, T. R. Auton, M. A. Firth, J. Li, R. A. Sykes, B. Waszkowycz, D. R. Westhead, and S. C. Young, *J. Comput.-Aided Mol. Design*, **11,** 193 (1997).
10. Y. Sun, T. J. A. Ewing, A. G. Skillman, and I. D. Kuntz (submitted).
11. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, *J. Mol. Biol.*, **161,** 269 (1982).
12. M. L. Connolly, *Science*, **221,** 709 (1983).
13. A. Feldman and L. Hodes, *J. Chem. Inf. Comput. Sci.*, **15,** 147 (1975).
14. *SYBYL, Version* 6.0.2, Tripos Associates, St. Louis, MO, 1993.
15. *SYBYL, Toolkit Manual*, Tripos Associates, St. Louis, MO, 1993.
16. C. Bron and J. Kerbosch, *Commun. ACM*, **16,** 575 (1973).
17. M. Rarey, B. Kramer, T. Lengauer, and G. Klebe, *J. Mol. Biol.*, **261,** 470 (1996).
18. W. Welch, J. Ruppert, and A. Jain, *Chem. Biol.*, **3,** 449 (1996).
19. W. Kabsch, *Acta Cryst.*, **A32,** 922 (1976).
20. W. Kabsch, *Acta Cryst.*, **A34,** 827 (1978).
21. D. R. Ferro and J. Hermans, *Acta Cryst.*, **A33,** 345 (1977).
22. The same algorithm is used in DOCK3.5 and DOCK4.0 to orient molecules in binding sites.
23. S. Makino and I. D. Kuntz, *J. Comput. Chem.*, **18,** 1812 (1997).
24. E. C. Taylor and D. J. Dumas, *J. Org. Chem.*, **46,** 1394 (1981).
25. A. Rosowsky, R. A. Forsch, H. Bader, and J. H. Freisheim, *J. Med. Chem.*, **34,** 1447 (1991).
26. A. Gangjee, F. Mavandadi, S. F. Queener, and J. J. McGuire, *J. Med. Chem.*, **38,** 2158 (1995).

27. A. Gangjee, N. Zaveri, M. Kothare, and S. F. Queener, *J. Med. Chem.*, **38,** 3660 (1995).
28. A. Gangjee, R. Devraj, and S. F. Queener, *J. Med. Chem.*, **40,** 470 (1997).
29. C. Bystroff, S. J. Oatley, and J. Kraut, *Biochemistry*, **29,** 3263 (1990).
30. Protein Data Bank, Chemistry Department, Building 555, Brookhaven National Laboratory, Upton, NY 11973.
31. M. Marsili, J. Gasteiger, and J. Croat, *Chim. Acta*, **52,** 601 (1980).
32. J. Gasteiger and M. Marsili, *Tetrahedron*, **36,** 3210 (1980).
33. S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, Jr., S. Profeta, and P. Weiner, *J. Am. Chem. Soc.*, **106,** 765 (1984).
34. S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, *J. Comput. Chem.*, **7,** 230 (1986).
35. T. J. A. Ewing and I. D. Kuntz, *J. Comput. Chem.*, **18,** 1175 (1997).
36. T. J. A. Ewing, S. Makino, A. G. Skillman, and I. D. Kuntz (submitted).